

Components in Experiment's Workflow Management Systems Infrastructure

Authors

Oliver Gutsche, Kenneth Herner, Burt Holzman, Tanya Levshina, Marco Mambelli, Parag Mhashilkar

Change Log

Version	Date	Change Description	Modified By
V0.1-V0.6	05/09/2016	Draft versions	Parag Mhashilkar
V1.0	05/13/2016	First Version	Parag Mhashilkar

1 Introduction

This document lists different components in a typical HEP Experiment's WMS infrastructure. It also tries to identify different services that provide the required functionality in case of ATLAS, CMS, FIFE and OSG VOs.

2 Components in HEP WMS Infrastructure

Every experiment labels components in their WMS infrastructure slightly differently. For consistency, this section defines the components that will be referenced in the rest of the document. In some systems the boundaries between these components are blurred and the same software does multiple functions or something in between.

2.1 Task

Task is a collection of jobs and is used as a main unit of computation. Task is defined and managed by users, e.g. the physicists running production or analysis.

In case of POMS used by FIFE, task is called as campaign. Sometimes a task is also referred as **Meta-Task** when it is composed of multiple sub-tasks (sequence/chain, parallel/bag, generic/graph).

2.2 Job

Data processing unit processed on a distributed system. Usually jobs are programmatically defined by splitting a task into smaller computational units. A typical job consists of following stages

- Input sandbox transfer
- Data processing
- Output stage out to MSS
- Output sandbox transfer

Failed jobs can be re-processed. Failures influence task completion time and its overall efficiency but do not necessarily cause a task failure.

2.3 Event

Smallest unit of raw scientific data typically processed by an experiment software. During its lifetime, a single job that runs experiment software can process one or more events.

In a traditional model, a job processes all the events in the input files before staging and storing back the results. However, in case of ATLAS, some job types also have event-level processing capability (see below) as these can retrieve and processes a single event or chunks independent from the file organization and store the results back. Storage, data movement systems and metadata catalog for the experiment data should also have additional functionality to support event level processing.

2.4 Task Bookkeeping

Component or service in a WMS infrastructure that is responsible for tracking individual tasks. Task bookkeeping may also act as a UI to the task submission system and/or the workflow manager.

2.5 Job Bookkeeping

Component or service in a WMS infrastructure that is responsible for tracking and managing individual jobs.

2.6 Event Bookkeeping

Component or service in a WMS infrastructure that is responsible for tracking individual events.

2.7 Task Submission System

Component or service used by a user to submit tasks to a task queue for processing.

2.8 Workflow Manager

Component or service responsible for managing tasks in the system.

2.9 Job Splitter

Component or service responsible for converting a task into one or more jobs. The splitting of a task can happen once (static) or be readjusted and customized to the resources available at the moment (dynamic).

2.10 Job Submission System

Component or service responsible for submitting jobs into a job processing queue. Queued jobs have to wait for computational resources to be available before they can be executed.

2.11 Resource Provisioning System

Service that executes a VO policy, and, on-demand, dynamically acquires resources. HEP and other scientific communities in the OSG use pilot-based systems as a resource provisioning system. ATLAS uses Panda while CMS, FIFE and OSG uses GlideinWMS for provisioning resources.

2.11.1 Pilot

Pilot is a job submitted by a resource provisioning system that runs in a worker node in the grid (or in a cloud virtual machine or in a HPC node). One of the critical functionalities of a pilot is to perform health checks and prepare the environment for jobs to run. After performing necessary tasks, the pilot advertises to the resource manager an available resource to run jobs. In distributed systems literature pilots are sometime referred as agents. Pilots provide to users a more uniform batch system interface to the resources acquired in grids, clouds and HPC clusters.

2.12 VO Resource Manager

Service responsible for matching queued jobs to the resources. These resources can be batch system resources or resources dynamically provisioned by a pilot.

2.12.1 Event level processing

The job matched with a resource could be also an event-service job. These jobs act as resource for the event-service: they request a chunk of events from the event-service manager, process them and store the results. It is an additional independent level of indirection (after pilots provision worker nodes from grids, clouds and HPC clusters) where some jobs running on top of pilots act as “pilot” for a finer grain resource manager dispatching smaller units of work, the events. Currently only ATLAS supports this.

3 WMS Components used by Experiments

	ATLAS	CMS	FIFE	OSG VO's
Experiment Software	Athena (AthenaMP, AthenaMT)	CMSSW	Various - Art + LarSoft - Gaudi [...]	Various Custom
Task Bookkeeping	DEFT	WMStats + ReqMgr ⁱ , CRAB Client+Server ⁱⁱ	POMS+SAM ⁱ , Custom	
Job Bookkeeping	Panda Server (JEDI)	WMStats ⁱ , CRAB Client+Server ⁱⁱ	POMS+Jobsub Client+SAM ⁱ , Jobsub Client (A)	HTCondor, Pegasus, Custom
Event Bookkeeping	Panda Server (JEDI/Event Service)	WMStats ⁱ , CRAB Client+Server ⁱⁱ		
Task Submission System	Panda Client (DEFT)	ReqMgr API ⁱ , CRAB Client+Server ⁱⁱ	POMS ⁱ , Jobsub Client, Custom	
Workflow Manager	Panda Server (JEDI)	WMAgent ⁱ , CRAB Server ⁱⁱ	POMS ⁱ , Custom, SAM based Custom	HTCondor, Pegasus, Custom
Job Splitter (Task -> n*Jobs)	Panda Server (JEDI)	WMAgent ⁱ , CRAB Server ⁱⁱ	Jobsub Server, Custom	HTCondor, Pegasus, Custom
Job Submission System (Jobs in queue)	None (Implicit in Panda Server)	WMAgent ⁱ , CRAB Server ⁱⁱ	Jobsub Server	HTCondor, Custom
Resource Provisioning System	Panda (Server + [APF Cron ssh])	GlideinWMS (Factory + Frontend)	GlideinWMS (Factory + Frontend)	GlideinWMS (Factory + Frontend)
VO Resource Manager (VO Pool/Matchmaker)	Panda Server	HTCondor	HTCondor	HTCondor

ⁱ Productionⁱⁱ Analysis

Resource Monitoring	Panda Monitoring	WLCG Dashboard, CMS Global Info from HTCondor	FIFEMon (Landscape)	
Site Availability Monitoring	SAM/ETF, HammerCloud, WLCG Site Readiness	SAM/ETF, HammerCloud, WLCG Site Readiness	FIFE Team Manual Operation	RSV
Job Monitoring	Panda Monitoring	WLCG Dashboard, WMStats ⁱ , CRAB Client ⁱⁱ	FIFEMon (Landscape), POMS ⁱⁱ Jobsub Client	HTCondor
Information Systems	AGIS	SiteDB, GlideinWMS (Factory config), WLCG Dashboard, Site Config (in CVMFS)	WMAgent (Resource Control) GlideinWMS (Factory Config)	OIM, GlideinWMS (Factory config)
Accounting	EGI Acct. Portal, Gratia	EGI Acct. Portal, Gratia	Gratia	Gratia
Data Cataloging	Rucio	PhEDEx + DBS	SAM	Custom
Data Movement	Rucio + WLCG-FTS3	PhEDEx + WLCG-FTS3 ⁱ , AOS+WLCG-FTS3 ⁱⁱ	SAM + ifdh	Custom

Table 1: WMS Components and Services used by HEP Experiments

4 FIFE Services Contact Members

Service	Contact
FIFEMon	Kevin Retzke
GlideinWMS	Parag Mhashilkar
Gratia	Tanya Levshina
IFDH	Marc Mengel
Jobsub	Dennis Box
POMS	Anna Mazzacane
SAM	Robert Illingworth

5 References

Panda Overview: <https://cd-docdb.fnal.gov:440/cgi-bin/ShowDocument?docid=5742>